

Data Collection

Most large organisations now use the online environment as a source for collecting and sharing data.

Examples include personal details or preferences, taking orders or providing credit card details.

Direct data collection

Organisations now rely on the customers to enter data directly to the organisation's database via a website.

This has a number of benefits for both the customer and the organisation.

Problems with input of incorrect data, eg. Incorrect email address

Why organisations collect data online?

Efficiency benefits:

- ▶ No double handling because direct data entry by customer (less staff required to enter data).
- ▶ Data can be prepopulated.
- ▶ Access at all hours.
- ▶ Instant, no post delay or phone queues

Why organisations collect data online?

Effectiveness benefits:

- ▶ Greater accuracy (no transcription errors or misread handwriting).
- ▶ Use of electronic validation (range checks, use of mandatory fields).
- ▶ Access to global markets.
- ▶ Marketing opportunities
- ▶ Provision of ongoing services (eg: emailed reminders or tracking of orders & transit).
- ▶ Also Expanded market base

Why customers supply data online?

- ▶ Convenience, eg. Housebound, time poor, individual preferences
- ▶ Variety of choice; global, compare prices, etc
- ▶ Reduces costs to consumer; no need to travel, free delivery
- ▶ Able to give feedback & opinions
- ▶ To exchange information, forums, wikis, blogs
- ▶ Immediate
- ▶ anonymity

Techniques used by organisations to acquire data online

- ▶ Online data acquisition
 - Data acquisition software: *To design appropriate user experiences (UX) to encourage users to interact with the system with ease by providing an interface that is both usable and accessible.*
- ▶ As users come from a variety of backgrounds so the design of the forms is vital.
- ▶ Appropriate user interface.
- ▶ Confirmation and acknowledgement is critical (eg: email message).
- ▶ Accessing an online database, intranet or extranet
- ▶ Web forum
- ▶ Online chat
- ▶ Multiple forms of contact, FAQ, email, phone
- ▶ Provide downloads

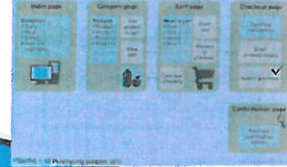
User flow diagrams

- ▶ A useful way to depict how a user will interact with a data collection tool.
- ▶ Shows the steps a user will use to complete a transaction.
- ▶ Allows for multiple entry points (website, search engine, online ad etc.).
- ▶ See page 11 for examples.

User flow diagrams



FIGURE 1.11 Job website UFD



Techniques for efficient & effective data collection

- ▶ Break large sets of questions into separate screens. Use clear labels.
- ▶ Use validation techniques.
- ▶ Use appropriate field types including check boxes, dropdown menus, input masks etc.)
- ▶ Be mobile friendly.
- ▶ Be user friendly.
- ▶ Prepopulate data

Data acquisition software

- ▶ PHP is a scripting language that creates dynamic webpage content. The webpage server accesses the database of information and searches for the information requested by the user. The user is in control of the info viewed per the request they provide – Google maps).
- ▶ ASP is also used to make webpages dynamic, primarily on the server side (Programs such as JavaScript are used on the client side).

Data acquisition software

- ▶ Back-end tools which gives statistics on who has visited a site and what information they accessed
- ▶ Cookies are small files which give data about the user to the website provider. They are sent to a computer hard drive that accesses a website. When the website is re-visited the browser will retrieve these details. They can be used as spyware to track people which can lead to security issues.

Protection of Rights

Most of the data collected by organisations needs to be kept secure. This is for a variety of reasons including:

- ▶ Competitive advantage.
- ▶ Damage or loss of data vital to functioning of organisation.
- ▶ To maintain public's confidence
- ▶ Privacy legislation.

Strategies to protect rights

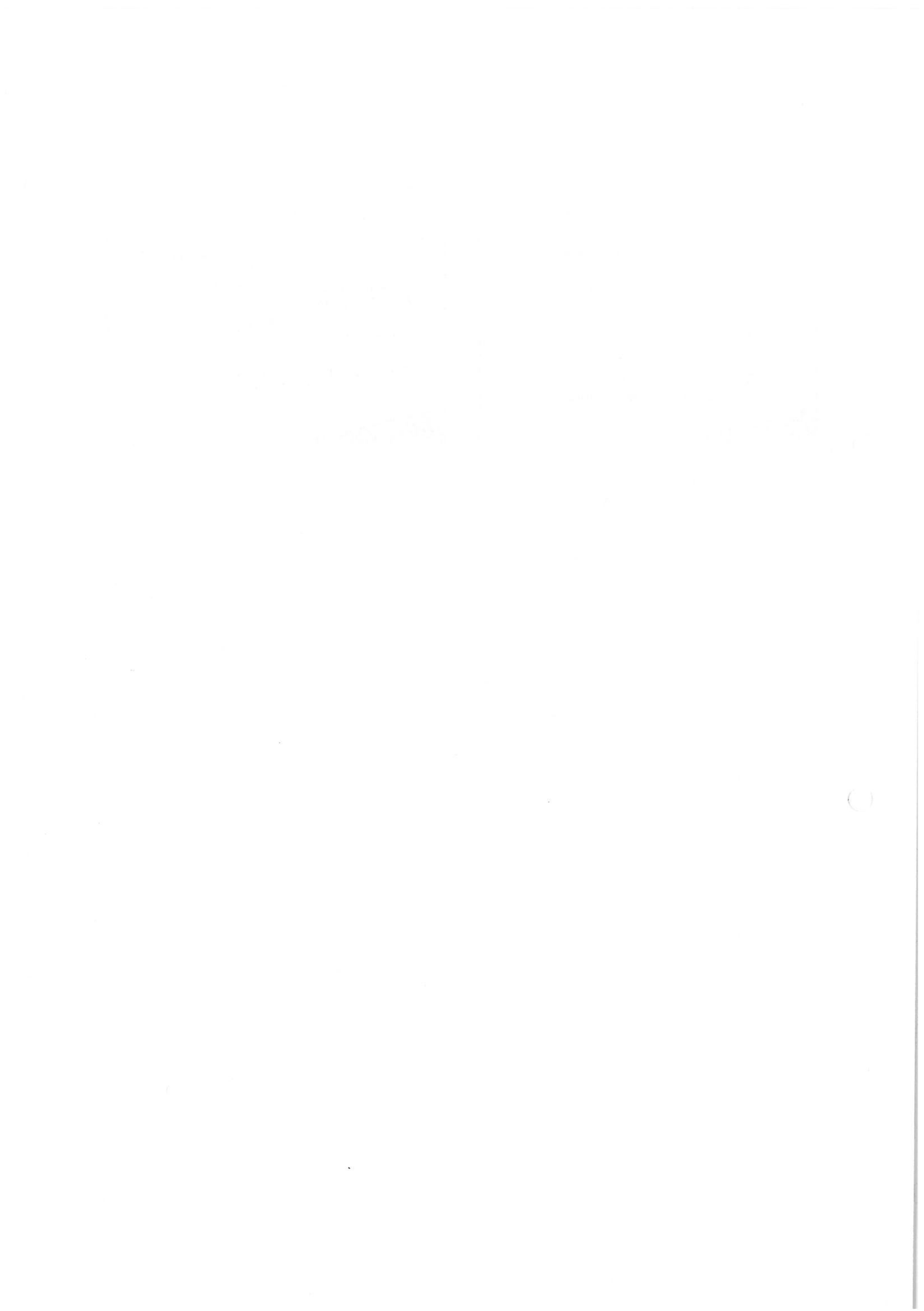
Guidelines can be put in place by organisations to enhance the protection of the information including:

- ▶ Security protocols (eg: TLS, SSL & HTTPS)
 - Protects data in transit
 - File encryption
 - Automatic timeout of idle connections
 - Using Captcha to deter robotic logins
 - Login using usernames & passwords

Strategies to protect rights

Guidelines can be put in place by organisations to enhance the protection of the information including:

- ▶ Privacy policies – should be displayed on website by law
 - Contacts link for people to use in case their rights have not been protected
- ▶ Shipping and returns policy – outlines to the customer how to return items



Design Principles

These are accepted characteristics that contribute to the **appearance** and **functionality** of solutions.

The objective of following these principles is to maximise the user experience through enhancing the efficiency and effectiveness of the information product being created.

Appearance

This refers to how the solution looks. Key principles include:

- Alignment
- Repetition
- Contrast
- Space
- Balance

Alignment

Refers to how we position objects in terms of creating a line for the eye to follow.

Examples include:

- Justifying text (usually left aligned).
- Aligning objects vertically and horizontally.
- See example on page 17 (Figure 1.22)

Repetition

Used to unite all the elements of a product by repeating patterns, textures and page elements.

Examples include:

- Using bullets to make a list.
- Making objects similar.
- Use symbols and links consistently.

Contrast

By placing objects dissimilar to each other beside each other they will stand out more.

Examples include:

- Light coloured background, dark text.
- Title banner contrasts with body of page.

Space

We can use space to separate on-screen objects. It allows them to stand out more with space around them.

Examples include:

- Use of “white space” to separate and group objects.
- Don’t overlap or overcrowd.

Balance

Trying to get balance over the whole page (screen).

Examples include:

- Symmetrical balance where objects are spread evenly over vertical and horizontal axis.
- Asymmetrical balance can also be achieved using objects of various weightings. See page 17 (Figure 1.23).

Functionality

This refers to how the solution works. Key principles include:

- Useability
- Flexibility and ease of use
- Accessibility

Useability

Refers to the solutions ability to be used and its availability to a user.

Examples include:

- Maximum compatibility with a wide range of devices.
- Provides feedback to the user.
- Support for recovery – robustness. Allows users to move back to change items.

Flexibility & ease of use

A solution should be as user friendly as possible for the widest audience possible.

Examples include:

- Allowing users some control over how they interact with the solution or how the data is presented (customisation).
- Making it easy for the user to understand how to perform tasks (intuitive).

Accessibility

The system is open to the widest range of users, including different locations, cultures, abilities and genders. Examples include:

- Ability to easily access the information.
- Simple and intuitive navigation.
- Allow for people with disabilities.
- Option to use different languages.
- Clear instructions.

Databases

A common software application used to develop information solutions is known as a database.

These applications are used to manage all types of data as they allow easy storing and manipulation.

A well known brand name database is Microsoft Access. We will be using this to create a solution for SAC 1.

Datatypes

Fields in databases hold specific type of data to ensure consistency in each field and assists validation

- 1 Text (string)
Mix of characters, alphanumeric, limit of 255
Names, etc, postcodes & telephone nos. text b/c they may contain spaces

Data types

- 2 Numeric - integer, floating point
 - Used when undertaking calculations
 - Integer, whole number, including negative nos; can't hold fractional parts; little storage needed
 - Decimal numbers, eg. dealing with financial transactions or percentages, floating point data type is used; currency data is a special type of floating point

Data types

- 3 Date
 - A variation of a numeric data type
 - Calculations can be performed on dates
- 4 Character
 - A text field only accepting a single alphanumeric character
- 5 Boolean
 - Used when data to be entered falls into one of two categories; very efficient storage
 - Usually True/False; Yes/NO OR On/Off

Structure of Databases

The common style of a database is to store data in tables consisting of rows and columns.

Each column is known as a field and each row contains a different record.

A simple example of a database is a phonebook. Fields include name, address and phone number, while each individual's details are a record.

Structure of databases

All databases are made up of data that is structured in a certain way.

- A **field** is a collection of characters that relates to a certain category.
- A **record** is a collection of fields that relate to one entity. Each record needs a unique identifier (primary key).
- A **table** is a collection of related records.

Structure of databases

The schools student database would be made up of the following components.

Student Name	← Field Name
Bill Stokes	← Field

Structure of databases

The schools student database would be made up of the following components.

Student Name	Year Level	House	Gender	
Bill Stokes	11	Merrylees	Female	← Record

Structure of databases

The schools student database would be made up of the following components.

Student Name	Year Level	House	Gender	
Bill Stokes	11	Merrylees	Female	← Table
Jim Dullard	12	Browne	Male	
Julie Zyroff	11	Browne	Female	
Leo Hogan	10	Nicholson	Male	
Jed Prince	9	Browne	Male	

Database Layouts

As table is designed to store data rather than enter or present data, Access uses other objects to perform these roles.

Forms: this allows the data to be entered in an efficient and effective manner. Its prime focus is to enter data quickly and to validate the data for accuracy.

Query: used when the user wants to select a set of data.

Database Layouts (continued)

As table is designed to store data rather than enter or present data, Access uses other objects to perform these roles.

Reports: this focuses on how the information is presented to the user. The emphasis is on effectiveness. A report often results from a query.

Macros: allow a set of predetermined tasks to be run (eg: run a query and print a report).

Structured Query Language (SQL)

Modern database systems use SQL (pronounced see kwel) to standardise how data is managed.

When creating a query in Access, the application uses SQL in the background to program the query, although the user may not be aware of this.

SQL has the advantage of being the standard for many database applications and therefore allows compatibility between different software.

Types of Databases

- **Flat file:** are simple style databases containing records and fields. The tables are not interlinked. This generally allows simple manipulation of data.
- **Relational:** this allows relationships between data in different tables to be stored. For this to occur both tables need to contain a common field. This allows a much more complex manipulation of data as well as reducing the amount of redundant (duplicate) data.

Databases - Flat File Vs Relational Database

Name	Department	Boss	Phone
Smith	Sales	Britney Lurgi	9123 456
Jones	Sales	Britney Lurgi	9123 456
Lennon	Sales	Britney Lurgi	9123 456
Sade	Transport	Tom Brick	9876 543
Masoch	Transport	Tom Brick	9876 543

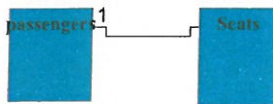
STAFF TABLE	
Name	Department
Smith	Sales
Jones	Sales
Lennon	Sales
Sade	Transport
Masoch	Transport

- + No unnecessary data repetition
- + Department info easily modified in one operation.

DEPARTMENTS TABLE		
Department	Boss	Phone
Sales	Britney Lurgi	9123 456
Transport	Tom Brick	9876 543

Types of Relationships, connection between data

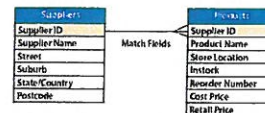
- One-to-one relationship:
Eg. Airline's passenger details:



Each passenger has only one seat and each seat can be assigned to only one passenger.
A record in one table is connected to only one record in a second table.

Types of Relationships, connection between data

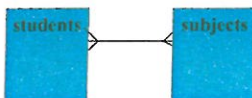
- One-to-many relationship:
Eg. customer details & job quotation:



The one supplier can provide MANY products, but each product comes from ONE supplier.
Also, eg. several workers share single telephone extensions; each extension record is related to several employee's records.

Types of Relationships, connection between data

- many-to-many relationship:
Eg. Student details and subjects tables:

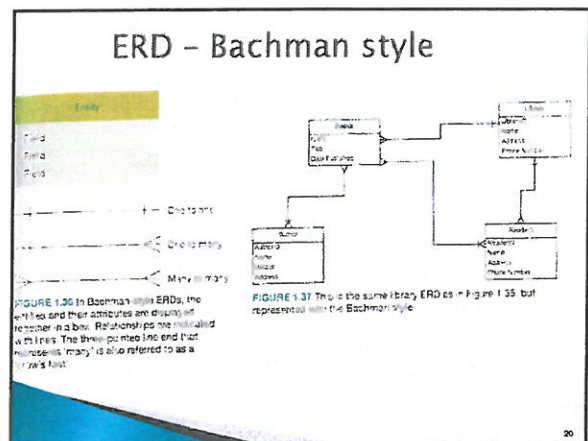
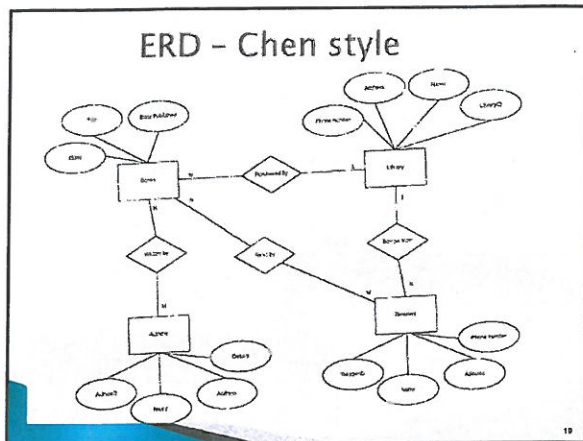


Each student studies many subjects and each subject has many students

ERD

- Used to establish the interrelationships between different data elements.
- An entity is a single person, place or thing about which data can be stored
- The characteristics, elements of data or attributes for each identity need to be determined
- Then relationships need to be determined:





Normalisation

Raw Data

Tournament Winners

<u>Tournament</u>	<u>Year</u>	<u>Winner</u>	<u>Winner Date of Birth</u>
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 September 1968
Des Moines Masters	1999	Al Fredrickson	21 July 1975
Indiana Invitational	1999	Chip Masterson	14 March 1977

1NF

First Name	Last Name	DOB	Tournament	Year
Chip	Masterton	14/03/1977	Indiana Invitational	1999
Al	Fredrickson	21/07/1975	Indiana Invitational	1998
Bob	Albertson	28/09/1968	Cleveland Open	1999
Al	Fredrickson	21/07/1975	Des Moines Masters	1999

- Data broken up into separate fields
- Date of birth converted into proper format

2NF

Tournament Winners

Player Code	First Name	Last Name	Tournament	Year
1	Chip	Masterton	Indiana Invitational	1999
2	Al	Fredrickson	Indiana Invitational	1998
3	Bob	Albertson	Cleveland Open	1999
2	Al	Fredrickson	Des Moines Masters	1999

Player Phone Numbers

Player Code	First Name	Last Name	DOB
1	Chip	Masterton	14/03/1977
2	Al	Fredrickson	21/07/1975
3	Bob	Albertson	28/09/1968

- Data grouped but ...
 - Data is still repeated

3NF

Tournaments

TournamentCode	Tournament
1	Indiana Invitational
2	Cleveland Open
3	Des Moines Masters

Tournament Winners

Player Code	TournamentCode	Year
1	1	1999
2	1	1998
2	3	1999
3	2	1999

Players

Player Code	First Name	Last Name	DOB
1	Chip	Masterton	14/03/1977
2	Al	Fredrickson	21/07/1975
3	Bob	Albertson	28/09/1968

- Data grouped meaningfully - Tournaments, Players, Winners
- No repeating data
- Codes used to link tables
- Relationships created

Normalise this data

Bounces Online Books

Name	Address	Book purchased	Item Cost	Date of purchase	Quantity	Total Cost
Tom Jones	56 Latrobe Street, Melbourne, VIC 3000	The Girl in the Hornet's Nest	\$24.95	08/03/2011	1	\$24.95
Tom Jones	65 Latrobe Street, Melbourne, VIC 3000	Curiosity Killed the Cat	\$14.95	08/03/2011	1	\$14.95
Mary Small	236 Smith Street, Collingwood VIC 3002	Lord of the Necklaces	\$18.95	10/03/2011	2	\$37.90
Mary Small	237 Smith Street, Collingwood VIC 3002	The Girl in the Hornet's Nest	\$24.95	10/03/2011	1	\$24.95
Fred Blogs	45 High Street, Sydney, NSW, 2000	The Hobby	\$13.95	12/03/2011	2	\$27.90
Fred Blogs	45 High Street, Sydney, NSW, 2000	Lord of the Necklaces	\$24.95	12/03/2011	1	\$24.95
Fred Blogs	45 High Street, Newcastle, NSW, 2000	The Girl in the Hornet's Nest	\$24.95	12/03/2011	1	\$24.95

First stage - 1NF

First Name	Last Name	Address1	Address2	Suburb	State	Postcode	Book purchased	Item Cost	Date of purchase	Quantity	Total Cost
Tom	Jones	56 Latrobe Street		Melbourne	VIC	3000	The Girl in the Hornet's Nest	\$24.95	08/03/2011	1	\$24.95
Tom	Jones	65 Latrobe Street		Melbourne	VIC	3000	Curiosity Killed the Cat	\$14.95	08/03/2011	1	\$14.95
Mary	Small	236 Smith Street		Collingwood	VIC	3002	Lord of the Necklaces	\$18.95	10/03/2011	2	\$37.90
Mary	Small	236 Smith Street		Collingwood	VIC	3002	The Girl in the Hornet's Nest	\$24.95	10/03/2011	1	\$24.95
Fred	Blogs	45 High Street		Sydney	NSW	2000	The Hobby	\$13.95	12/03/2011	2	\$27.90
Fred	Blogs	45 High Street		Sydney	NSW	2000	Lord of the Necklaces	\$24.95	12/03/2011	1	\$24.95
Fred	Blogs	45 High Street		Sydney	NSW	2000	The Girl in the Hornet's Nest	\$24.95	12/03/2011	1	\$24.95

Second Stage – 2NF

Customer table

CustomerCode	First Name	Last Name	Address1	Address2	Suburb	State	Postcode
116	Tom	Jones	56 Latrobe Street		Melbourne	VIC	3000
457	Mary	Small	236 Smith Street		Collingwood	VIC	3002
890	Fred	Blogs	45 High Street		Sydney	NSW	2000

Books Purchased table

CustomerCode	Book purchased	Item Cost	Date of purchase	Quantity	Total Cost
116	The Girl in the Hornet's Nest	\$24.95	08/03/2011	1	\$24.95
116	Curiosity Killed the Cat	\$14.95	08/03/2011	1	\$14.95
457	Lord of the Necklaces	\$18.95	10/03/2011	2	\$37.90
457	The Girl in the Hornet's Nest	\$24.95	10/03/2011	1	\$24.95
890	The Hobby	\$13.95	12/03/2011	2	\$27.90
890	Lord of the Necklaces	\$24.95	12/03/2011	1	\$24.95
890	The Girl in the Hornet's Nest	\$24.95	12/03/2011	1	\$24.95

Third Stage - 3NF

Customer Table

CustomerCode	First Name	Last Name	Address1	Address2	Suburb	State	Postcode
116	Tom	Jones	56 Latrobe Street		Melbourne	VIC	3000
457	Mary	Small	236 Smith Street		Collingwood	VIC	3002
890	Fred	Blogs	45 High Street		Sydney	NSW	2000

Purchases Table

CustomerCode	BookCode	Date of purchase	Quantity	Total
116	1	08/03/2011	1	\$24.95
116	15	08/03/2011	1	\$14.95
457	36	10/03/2011	2	\$37.90
457	1	10/03/2011	1	\$24.95
890	4	12/03/2011	2	\$27.95
890	36	12/03/2011	1	\$28.95
890	1	12/03/2011	1	\$24.95

Books Table

BookCode	Book Name	Genre	Item Cost
1	The Girl in the Hornet's Nest	Murder Mystery	\$24.95
15	Curiosity Killed the Cat	Romance	\$14.95
36	Lord of the Necklaces	Fantasy	\$18.95
4	The Hobby	Fantasy	\$13.95



1.3 b
VIA KB
edited
Feb 2016

From
Paul Posner
St Francis Xavier CMA
Beaconsfield

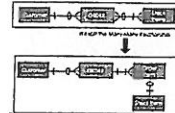
3/02/2016

DETERMINING A RDBMS STRUCTURE (Data Normalisation)

Informatics

What is RDBMS ?

- The acronym "RDBMS" stands for:
Relationship Database Management System
 - which is the full technical name for a Database.
- When we have far too much data to handle in a Spreadsheet, we need to use a Database.
- Eg. Online Shop, School Information, Car Registrations, Police Crimes, Medicare, Taxes, etc.
- Database has Tables containing Fields
 - Book has Chapters containing Pages
 - USB Stick or SD card has Folders containing Files
 - Chest of Drawers has separate sock drawer, shirts drawer, etc
 - A stack of Unit 3 and 4 folders has topics for Y12 subjects.



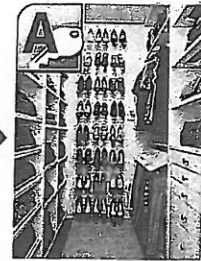
Planning a Database

- Creating a database takes some thought. We must first plan the structure, get it right, then start entering the data.
- It is a bit like building a house – you don't want to have to tear down a half built house, because you got the foundations and Frame wrong, and now it is all "wonky".
- So before we get to the input of data stage, (Development Stage of the PSM) we need to make sure that we get the structure and design as close to perfect as possible.



Planning a Database

- Designing a database is a lot like getting your room organised, so that you can find things easily, and add new things to the right location.



What is RDBMS Structure ?

- Once we have identified that a relational database is required to solve a problem, we need to establish how to break up the data that will be supplied by the client into fields and tables.
- It is extremely important to plan this carefully in order to maximise the efficiency of a relational database structure.
- Once the fields and tables have been decided upon, the next step will be to apply the table Normalisation Rules to ensure that there are no fields that are not needed, or unnecessarily duplicated, and that the "integrity" (organisation and accuracy) of the data is maintained.



What is RDBMS Structure ?

- An Online Auction Site, like a simplified eBay, needs many different Tables in its Database. The Table setup needs to be well designed.
- We cannot just make up a big spreadsheet for each Customer!



What is Table Normalisation?



- Normalising the tables will need to occur after you breakdown the relevant data into fields and tables.
- It also relates to organising the data in a relational database so that :-
 - Data **repetition** is minimised
 - Data **access** is maximised
- The normalisation rules are intended to give you as the designer of the database, a tool to ensure that data integrity is maintained.
- Data Integrity** ensures that all data in a database is complete, and is sufficiently broken down to avoid duplication and repeats.

Why Normalise?



- Removing data **repetition** saves lots of storage space, speeds up data access and reduces errors.
- Changes** need only be made in one place rather than in many places. (Eg. Change Address only in Student Table, not also in Fees Table, and Reports Table).
- More **powerful** data access is possible when fields are broken down properly. (Eg. Address has Street, Suburb and Postcode as separate fields)

Eg. Find what Postcodes have your biggest customers for business location and/or advertising and marketing purposes.



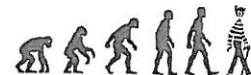
Why Normalise?

- Allows more information to be easily **Stored and Reported**
- Allows users to get all sorts of **Information** out of the stored data i.e. How many products did we sell each month?

Compare with Last Month by Categories

Category Name	Current Month		Last Month	
	Quantity	Amount	Quantity	Amount
Beverages	925	\$27,761.08	834	\$24,529.15
Candies	278	\$10,773.37	283	\$8,283.87
Cereals	860	\$22,877.18	475	\$10,874.08
Dairy Products	581	\$17,885.33	508	\$11,454.59
Grains/Cereals	109	\$3,325.40	210	\$4,000.01
Meat/Poultry	82	\$4,063.66	344	\$23,324.86
Produce	351	\$12,031.23	37	\$1,172.80
Seafood	609	\$9,216.56	594	\$12,531.12
Total	4,065	\$104,824.51	3,288	\$103,519.49

The 6 "Normal" Forms



- There are 6 normal forms and each rule is applied successively from the first normal form (1NF) to the sixth normal form (6NF). Although only the first three are used for the majority of databases.
- If you have planned your database well, it will automatically comply with the normalisation rules.
- The rules are actual guidelines (not laws) for the actual structuring of database tables and fields.

Getting to Third Normal Form

- REMEMBER** – 1st and 2nd normal forms are stages/steps to achieving the objective, which is 3rd normal form.



☒ STEP ONE
☒ STEP TWO
☒ STEP THREE



The Normal Forms



- First Normal Form (1NF) – fields split up properly into individual component parts
- Second Normal Form (2NF) – first stage of breaking up the data into meaningful groupings called tables, (Everything in a Table is relevant for that Table)
- Third Normal Form (3NF) – data completely broken up into tables and linked by ID Codes / Keys

1NF - First Normal Form



- The 1NF guidelines are basically common sense.

1. Eliminate duplicate data where possible
2. Break up fields so only one data item is in each field
3. Convert data into correct format
4. Start to organise the data into meaningful groupings

Your turn... attempt to repair this table using 1NF

CustomerID	Name	Phone
111	Fred Smith	024567 8900
222	Mary Jones	024567 8900
333	Sam Jones	024567 8900

1. Eliminate duplicate data where possible
2. Break up fields so only one data item is in each field
3. Convert data into correct format
4. Start to organise the data into meaningful groupings

Repaired using 1NF rules



CustomerID	FirstName	LastName	PhoneNbr
111	Fred	Smith	02 366 3456
222	Mary	Jones	024567 8900
333	Sam	Jones	024567 8900

- This table now allows customers to be sorted and searched by first name and/or surname separately.
- Also, the names can be used individually, like "Dear Fred" instead of "Dear Fred Smith"

Your Turn: Repair This using 1NF rules

CustomerID	Address
111	366 Fred St, Sale, 3190
222	2/45 Richmond Lane, Richmond, 3121
333	135 Spring St, Melbourne, 3000

1. Eliminate duplicate data where possible
2. Break up fields so only one data item is in each field
3. Convert data into correct format
4. Start to organise the data into meaningful groupings

Hint:

- An address like "3 Fred St, Sale, 3586" has 3 pieces of data: street address, town, and postcode.

Repaired!



Customer ID	Street	Suburb	Postcode
111	366 Fred St	Sale	3190
222	2/45 Richmond Lane	Richmond	3121
333	135 Spring St	Melbourne	3000

- Now each field can be searched & sorted and used individually (e.g. addressing envelopes, or finding the suburbs most customers live in for an upcoming Marketing Campaign)

Your Turn: Repair This using 1NF rules

Student Table:

Student	Age	Subject
Adam	15	Biology, Maths
Alex	14	Maths
Sam	17	Maths

1. Eliminate duplicate data where possible
2. Break up fields so only one data item is in each field
3. Convert data into correct format
4. Start to organise the data into meaningful groupings

Note:

- It is okay for Maths to be duplicated, as several different students are studying it.

Repaired!

In First Normal Form, any row must not have a column in which more than one value is saved, like separated with commas. Rather than that, we must separate such data into multiple rows.

Student Table following 1NF will be :

Student	Age	Subject
Adam	15	Biology
Adam	15	Maths
Alex	14	Maths
Stuart	17	Maths

We need to store each subject for a particular student as a separate record, not all bunched up in one column as multiple items.

Are these Tables in 1NF ?

1. Eliminate duplicate data where possible
2. Break up fields so only one data item is in each field
3. Convert data into correct format
4. Start to organise the data into meaningful groupings

1.

ID	Firstname	Surname	Full Name	Address	City	Postcode
10	Tom	Smith	Tom Smith	12 112 Street	London	W1E 1JG17

2.

ID	IP Address	Username	Last accessed	Access	Status	Active
1000	198.168.1.5	Smith	2000/10/21 14:10	Save file	Success	Y

3.

ItemID	Product	Description	Price	Weight	Colour	Material
234	Shoe	High Heel	15	Red	Blue	Brown

4.

StudentID	Firstname	Surname	Age	Subject	Teacher
1354	Tom	Smith	15	Maths	Mr

ANSWERS: Are these Tables in 1NF ?

1. Table 1: Not in 1NF because "Full Name" is duplicate data
2. Table 2 is in 1NF (No Duplicates, Data items broken down ok)
3. Table 3 is Not in 1NF - Duplicate Colour Fields all named the same
4. Table 4 is in 1NF (No Duplicates, Data items broken down ok)

1.

ID	Firstname	Surname	Full Name	Address	City	Postcode
10	Tom	Smith	Tom Smith	12 112 Street	London	W1E 1JG17

2.

ID	IP Address	Username	Last accessed	Access	Status	Active
1000	198.168.1.5	Smith	2000/10/21 14:10	Save file	Success	Y

3.

ItemID	Product	Description	Price	Weight	Colour	Material
234	Shoe	High Heel	15	Red	Blue	Brown

4.

StudentID	Firstname	Surname	Age	Subject	Teacher
1354	Tom	Smith	15	Maths	Mr

2NF – Second Normal Form

- Achieving 2NF means 1NF has already been achieved
- Each normal form builds on the previous forms
- Removes more duplicate data, and moves this data to other tables.
- Deals with design problems that could threaten **data integrity**.

2NF – Second Normal Form

- Remove subsets of data that apply to multiple rows of a table and places them in separate tables.
- Creates relationships between these new tables and their predecessors using unique keys.
- Look at Tables 1.4 to 1.9 of your textbook which shows how 2NF has been used to create separate tables, **where everything in each table is fully dependant on the Primary Key**

2NF – Second Normal Form

Here is our previous 1NF Table:

Student Table following 1NF will be :

Student	Age	Subject
Adam	15	Biology
Adam	15	Maths
Alex	14	Maths
Stuart	17	Maths

This Table is NOT in Second Normal Form

The Problem here is that we store a student's Age many times, by putting it in with each subject they do. (Eg. Adam who is 15 years old) What we really need is to have a Separate Table to store Age.

2NF Repaired!

New Student Table following 2NF will be:

Student	Age
Adam	15
Jake	14
Stuart	17

Age Table

New Subject Table introduced for 2NF will be:

Student	Subject
Adam	Biology
Adam	Maths
Adam	Maths
Stuart	Maths

Subjects Table

- We now have two separate tables where data items are not unnecessarily duplicated for any Student.

2NF – Second Normal Form

Here is a Table About Concerts:

CONCERT

Venue	Artist	Attendance	Price	Style
Merbury	Guns About	50000	12334	Guitarband
SEI	Guns About	15000	56433	Female band

The table above is using both the venue and artist as the compound primary key.

It is in this situation that the extra rule for second normal form comes in handy. The rule states

- Non-key attributes must depend on every part of the primary key

This Table is NOT in Second Normal Form

The Problem here is that the "Style" of the Performer is not totally relevant to how many people attended and how many \$ were made. **What we really need is to have a Separate Table to store Styles.**

2NF Repaired!

Not in 2NF!

CONCERT

Venue	Artist	Attendance	Price	Style
Merbury	Guns About	50000	12334	Guitarband
SEI	Guns About	15000	56433	Female band
SEI	Guns About	15000	56433	Guitarband

This table needs to be split so that non-dependent attributes are removed and only stored once.

In this case a 'style' table is formed that has Artist as the simple primary key

CONCERT

Venue	Artist	Attendance	Price
Merbury	Guns About	50000	12334
SEI	Guns About	15000	56433
SEI	Guns About	15000	56433

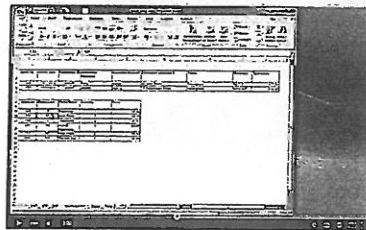
STYLE

Artist	Style
Guns About	Guitarband
Female band	Female band

Now the rule for 2NF is being met by both tables - every non-key attribute is depending on the complete primary key. There is no redundant data.

2NF – Second Normal Form

2NF not related items duplication can be tricky as shown in this video:



<https://www.youtube.com/watch?v=8PwomfwMMYQ>

3NF – Third Normal Form

Third normal form (3NF) goes one step further:

- Uses ID codes and Number IDs to minimize the amount of storage
- Uses these codes as links to other tables so we can find any information we need
- Sets up relationships between tables
- In each table we must have fields that are only dependant on the primary key
- Also separates data into reference and transaction data.

Normalise this data using the rules 1NF, 2NF & 3NF

1st

2nd

3rd

Source	Online	Books					
Tom	36 Ashburn Street, Melbourne VIC 3000	The Girl in the Hornet's Nest	\$24.95	06/03/2011	1		\$24.95
Tom	65 London Street, Melbourne VIC 3000	Curiosity Killed the Cat	\$14.95	06/03/2011	1		\$14.95
May	236 Smith Street, Collingwood VIC 3062	Lord of the Necklaces	\$19.95	10/03/2011	2		\$39.90
May	236 Smith Street, Collingwood VIC 3062	The Girl in the Hornet's Nest	\$24.95	10/03/2011	1		\$24.95
Fred	45 High Street, Sydney, NSW 2000	The Hobbit	\$13.95	12/03/2011	2		\$27.90
Fred	45 High Street, Sydney, NSW 2000	Lord of the Necklaces	\$24.95	12/03/2011	1		\$24.95
Fred	45 High Street, Newcastle, NSW 2000	The Girl in the Hornet's Nest	\$24.95	12/03/2011	1		\$24.95

First stage - 1NF

1st

Customer ID	Customer Name	Address	City	State	Postcode	Book Title	Author	Date	Quantity	Price
116	Tom Jones	56 Latrobe Street	Melbourne	VIC	3000	The Girl in the Homel's Nest		08/03/2011	1	\$24.95
116	Tom Jones	56 Latrobe Street	Melbourne	VIC	3000	Cruisely Killed the Cat		08/03/2011	1	\$14.95
457	Mary Small	236 Smith Street	Collingwood	VIC	3002	Lord of the Necklaces		10/03/2011	2	\$37.90
457	Mary Small	236 Smith Street	Collingwood	VIC	3002	The Girl in the Homel's Nest		10/03/2011	1	\$24.95
890	Fred Blogs	45 High Street	Sydney	NSW	2000	The Hobby		12/03/2011	2	\$27.90
890	Fred Blogs	45 High Street	Sydney	NSW	2000	Lord of the Necklaces		12/03/2011	1	\$18.95
890	Fred Blogs	45 High Street	Sydney	NSW	2000	The Girl in the Homel's Nest		12/03/2011	1	\$24.95

Second Stage - 2NF

2nd

The 1NF Data is about Customers and Books Purchased.
It needs to be broken into two tables:
Customer Details and Books Purchased.

There are also other problems such as customers having identical names.

1. Customer Details Table - Introduce an ID Number Code (3NF concept) to deal with Duplicate Customer names.

Customer ID	First Name	Last Name	Address	City	State	Postcode
116	Tom	Jones	56 Latrobe Street	Melbourne	VIC	3000
117	Tom	Jones	45 London Street	Melbourne	VIC	3000
457	Mary	Small	236 Smith Street	Collingwood	VIC	3002
890	Fred	Blogs	45 High Street	Sydney	NSW	2000

See next Slide for Books Purchased Table

Second Stage - 2NF

2nd

2. Books Purchased Table

Customer ID	Book Title	Author	Date	Quantity	Price
116	The Girl in the Homel's Nest		08/03/2011	1	\$24.95
116	Cruisely Killed the Cat		08/03/2011	1	\$14.95
457	Lord of the Necklaces		10/03/2011	2	\$37.90
457	The Girl in the Homel's Nest		10/03/2011	1	\$24.95
890	The Hobby		12/03/2011	2	\$27.90
890	Lord of the Necklaces		12/03/2011	1	\$18.95
890	The Girl in the Homel's Nest		12/03/2011	1	\$24.95

Third Stage - 3NF

3rd

Customer ID	First Name	Last Name	Address	City	State	Postcode
116	Tom	Jones	56 Latrobe Street	Melbourne	VIC	3000
457	Mary	Small	236 Smith Street	Collingwood	VIC	3002
890	Fred	Blogs	45 High Street	Sydney	NSW	2000

Note that we do not store the Total because it can be calculated as needed by looking up the Price in the Books Table.

Customer ID	Book Title	Author	Date	Quantity	Price	Total
116	The Girl in the Homel's Nest		08/03/2011	1	\$24.95	\$24.95
116	Cruisely Killed the Cat		08/03/2011	1	\$14.95	\$14.95
457	Lord of the Necklaces		10/03/2011	2	\$18.95	\$37.90
457	The Girl in the Homel's Nest		10/03/2011	1	\$24.95	\$24.95
890	The Hobby		12/03/2011	2	\$13.95	\$27.90
890	Lord of the Necklaces		12/03/2011	1	\$18.95	\$18.95
890	The Girl in the Homel's Nest		12/03/2011	1	\$24.95	\$24.95

Book Title	Author	Genre	Price
The Girl in the Homel's Nest		Mystery	\$24.95
Cruisely Killed the Cat		Romance	\$14.95
Lord of the Necklaces		Fantasy	\$18.95
The Hobby		Fantasy	\$13.95

Benefits of Normalisation



1. The database does not have redundant data, it is smaller in size so less money needs to be spent on storage
2. Because there is less data to search through, it is much faster to run a query on the data
3. Because there is no data duplication there is better data integrity and less risk of mistakes.
4. Because there is no data duplication there is less chance of incorrectly storing two or more different copies of the data
5. A change or Update can be made which can then instantly be cascaded across to any related records.

Problems with Normalisation



1. You need to be careful with trying to make data atomic. Just because you can split some types of data further, it isn't always necessarily the correct thing to do.
 - For example, telephone number might contain the area code "03" followed by the number 97026380. It wouldn't be sensible to separate out these two items.
2. You can end up with many more tables to deal with than in an un-normalised database.
3. The more tables and the more complex the database, the slower queries can be to run
4. It is necessary to assign more relationships to interact with the larger numbers of tables
5. With more tables, setting up queries can become more complex

Glossary Terms to Know

- **Table**
- **Fields**
- **Records**
- **Field types** – including Boolean
- **Validation**
 - range, type, existence, existence in limited list
- **Key field, primary key, match field**
- **Foreign key**
- **Relational**
- **Flat-file**
- **Relationships**
 - 1:1
 - 1:many
 - many:many



Exam Knowledge



- Students are required to be able to answer the following questions: -
 1. Define data integrity.
 2. Explain the difference between all 3 forms of normalisation?
 3. Explain why normalisation is important?
 4. Name the 3 different forms of relationships?
 5. Describe what is a primary key and how is it different to a foreign key?
- Students are also required to normalise to 3NF using data provided.

Acknowledgement

- This presentation was made with the help of the following website, which I encourage you to use in preparing yourself to better understand "Normalisation"
- http://www.teach-ict.com/as_as_computing/ocr/H447/F453/3_3_9/normalisation/miniweb/index.htm

Information from various Edulists Powerpoints has also been included





A Normalisation Example



Mark Kelly
McKinnon Secondary College
Vceit.com
Based on work by Robert Timmer-Arends

Take the following table.

StudentID is the primary key.

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	\$50	B
					Maths	\$50	A
					Info Tech	\$100	B+

Is it 1NF?

No. There are repeating groups (subject, subjectcost, grade)

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	\$50	B
					Maths	\$50	A
					Info Tech	\$100	B+

How can you make it 1NF?

Create new rows so each cell contains only one value

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	\$50	B
					Maths	\$50	A
					Info Tech	\$100	B+



StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	\$50	B
19594332X	Mary Watson	10 Charles Street	Bob	Red	Maths	\$50	A
19594332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	\$100	B+

But now look – is the *studentID* primary key still valid?

No – the studentID no longer uniquely identifies each row

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19394332X	Mary Watson	10 Charles Street	Bob	Red	English	\$50	B
19394332X	Mary Watson	10 Charles Street	Bob	Red	Maths	\$50	A
19394332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	\$100	B-

You now need to declare *studentID* and *subject* together to uniquely identify each row.

So the new key is StudentID and Subject.

So. We now have 1NF.

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19394332X	Mary Watson	10 Charles Street	Bob	Red	English	\$50	B
19394332X	Mary Watson	10 Charles Street	Bob	Red	Maths	\$50	A
19394332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	\$100	B-

Is it 2NF?

Studentname and address are dependent on studentID (which is part of the key)
This is good.

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19394332X	Mary Watson	10 Charles Street	Bob	Red	English	\$50	B
19394332X	Mary Watson	10 Charles Street	Bob	Red	Maths	\$50	A
19394332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	\$100	B-

But they are **not** dependent on *Subject* (the *other* part of the key)

And 2NF requires...

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19394332X	Mary Watson	10 Charles Street	Bob	Red	English	\$50	B
19394332X	Mary Watson	10 Charles Street	Bob	Red	Maths	\$50	A
19394332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	\$100	B-

All non-key fields are dependent on the ENTIRE key (studentID + subject)

So it's not 2NF

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19394332X	Mary Watson	10 Charles Street	Bob	Red	English	\$50	B
19394332X	Mary Watson	10 Charles Street	Bob	Red	Maths	\$50	A
19394332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	\$100	B-

How can we fix it?

Make new tables

- Make a new table for each primary key field
- Give each new table its own primary key
- Move columns from the original table to the new table that matches their primary key...

Step 1

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19394332X	Mary Watson	10 Charles Street	Bob	Red	English	\$50	B
19394332X	Mary Watson	10 Charles Street	Bob	Red	Maths	\$50	A
19394332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	\$100	B-

STUDENT TABLE (key = StudentID)

Step 2

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19394332X	Mary Watson	10 Charles Street	Bob	Red	English	\$50	B
19394332X	Mary Watson	10 Charles Street	Bob	Red	Maths	\$50	A
19394332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	\$100	B-

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19394332X	Mary Watson	10 Charles Street	Bob	Red

SUBJECTS TABLE (key = Subject)

Step 3

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	\$50	B
19594332X	Mary Watson	10 Charles Street	Bob	Red	Maths	\$50	A
19594332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	\$100	B-

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	\$50
Maths	\$50
Info Tech	\$100

RESULTS TABLE (key = StudentID+Subject)

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	\$50	B
19594332X	Mary Watson	10 Charles Street	Bob	Red	Maths	\$50	A
19594332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	\$100	B-

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	\$50
Maths	\$50
Info Tech	\$100

RESULTS TABLE (key = StudentID+Subject)

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B-

Step 4 - relationships

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	\$50
Maths	\$50
Info Tech	\$100

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B-

RESULTS TABLE (key = StudentID+Subject)

Step 4 - cardinality

STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

1 Each student can only appear
ONCE in the student table

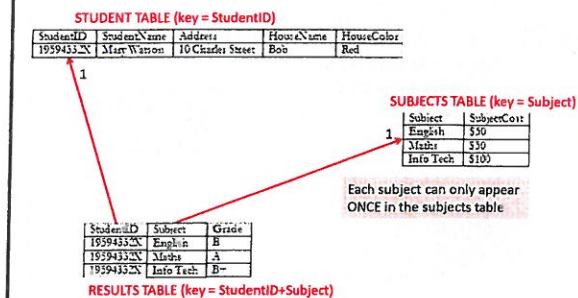
SUBJECTS TABLE (key = Subject)

Subject	SubjectCost
English	\$50
Maths	\$50
Info Tech	\$100

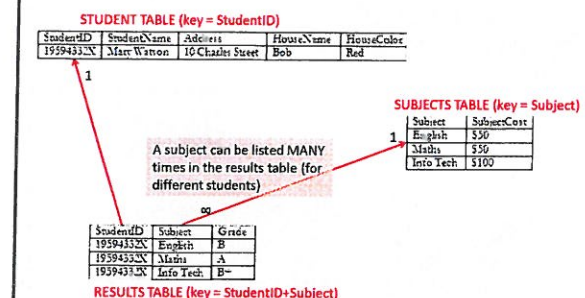
StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B-

RESULTS TABLE (key = StudentID+Subject)

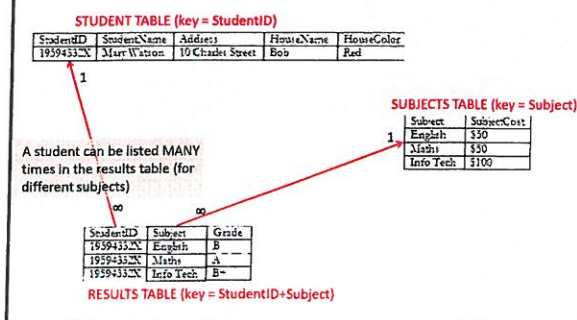
Step 4 - cardinality



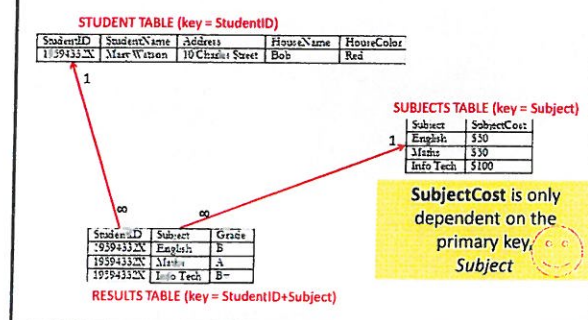
Step 4 - cardinality



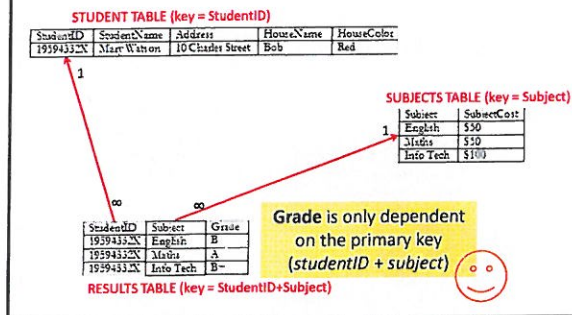
Step 4 - cardinality



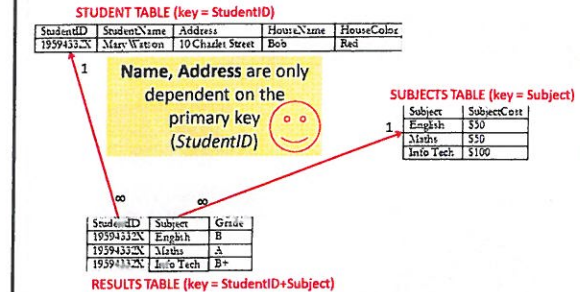
A 2NF check



A 2NF check



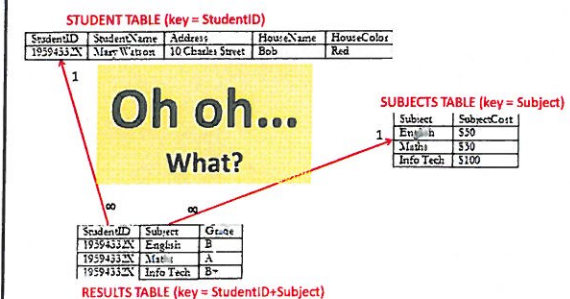
A 2NF check



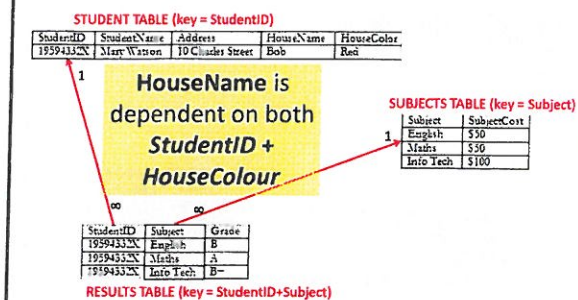
So it is
2NF!

But is it 3NF?

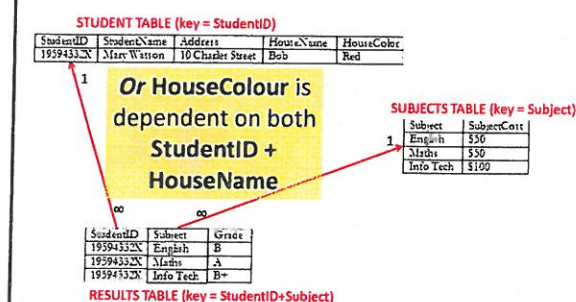
A 3NF check



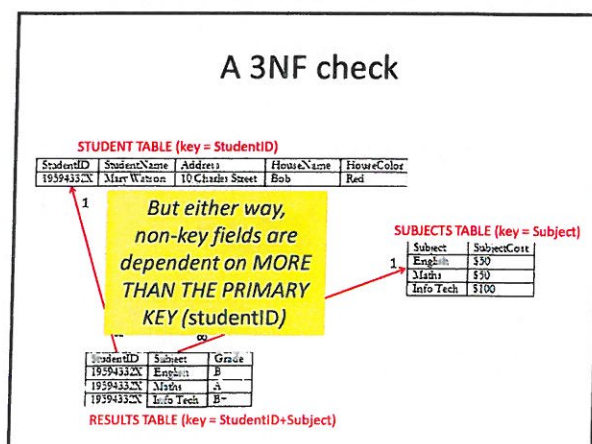
A 3NF check



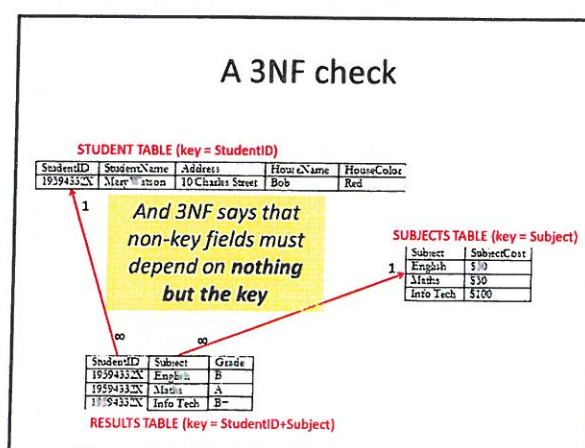
A 3NF check



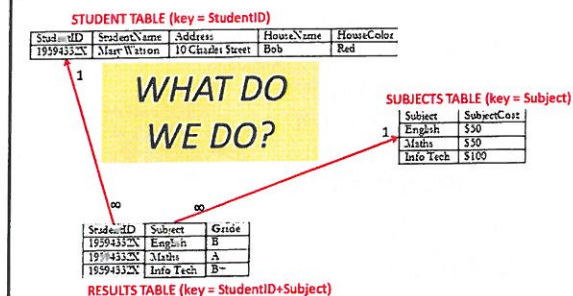
A 3NF check



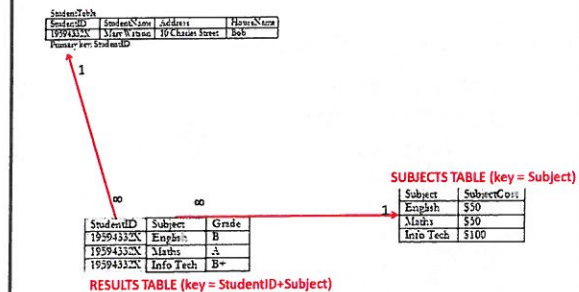
A 3NF check



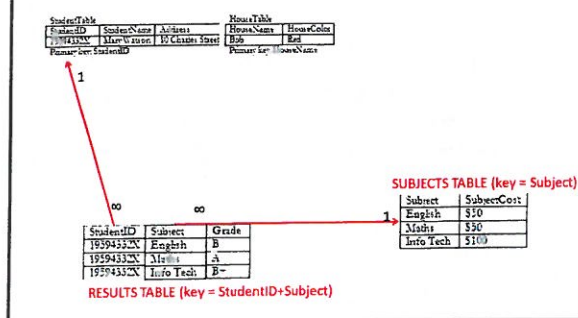
A 3NF check



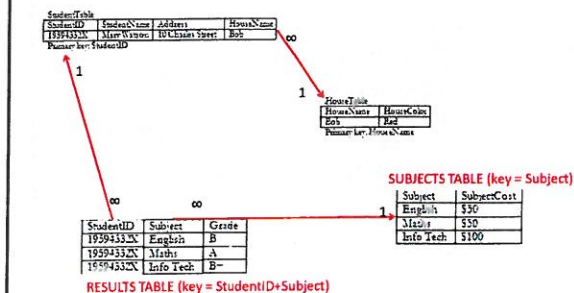
Again, carve off the offending fields



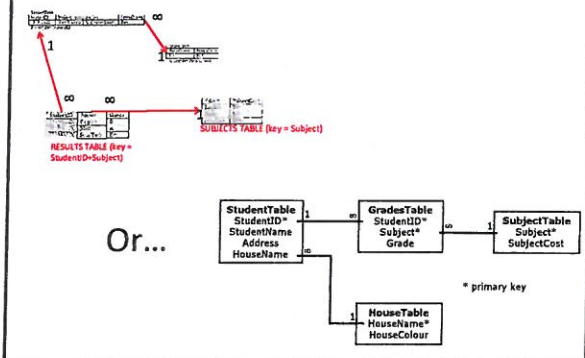
A 3NF fix



A 3NF fix



A 3NF win!

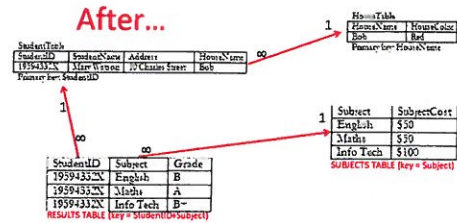


The Reveal

Before...

StudentID	StudentName	Address	HouseName	HouseColour	Subject	SubjectCost	Grade
19594352N	Mr. Watson	10 Clinton Street	Bob	Red	English	\$30	B
					Maths	\$30	A
					Info Tech	\$100	B-

After...





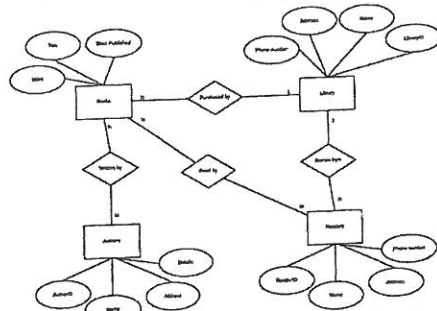
Designing a Database

We can use a variety of tools to plan the structure and appearance of a database. The design would usually cover the following areas:

- **Representing Structure:** data dictionaries, data structure diagrams, flowcharts, IPO charts and query diagrams.
- **Relationships:** ERD's & structure charts.
- **Appearance:** layout diagrams & mock-ups

Design Tools

- An **Entity-relationship diagram (ERD)** is used to establish the inter-relationships between different data elements (p 28)



Naming Conventions

At the design stage it is also vital that every field and object (form, query, report etc.) are properly named.

This means that they are easily identified and do not cause confusion, especially if multiple developers (or a team) are working on the solution.

An example of effective naming for fields is to prefix those in a certain table so they can be identified as belonging to that table.

Spaces and underscoring should be avoided in Access (use CamelCase).

Design Tools

- A **data structure diagram** illustrates the relationships and links between tables. (p 39)

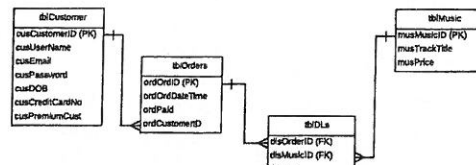


FIGURE 1.44 An example of a data structure diagram with one-to-many relationships. This DSD represents the database table relationships shown in Figure 1.45.

Design Tools

- A **data dictionary** explains how each field should be set up in a table. (p 37)

TABLE 1.21 The tblCustomers table for POTG

Field	Display	Field Size	Field Mask	Column Name	Description	Validation Rule	Validation Text
cusCustomerID (PK)	Text	4		Customer ID	Customer's unique assigned code	Between 200 and 9999	Customer ID number must be between 2000 and 9999
cusFullName	Text	20		Full Name	Customer's full name		
cusGreenName	Text	20		Green Name	Customer's green name		
cusDelAddress	Text	40		Address	Customer's delivery address		
cusDelSuburb	Text	15		Suburb	Customer's delivery suburb		
cusDelPostcode	Text	4		Post Code	Customer's delivery post code	Between 2020 and 9999	Values must be between 2020 and 9999 inclusive
cusMobile	Text	12	9999 9999	Mobile	Customer's mobile number		
cusEmail	Text	40		Email	Customer's email address		
cusIsSub	Boolean	1		Is Sub	Is customer a member of the Pizza Club?		

Design Tools

- A **query design** specifies what each query (find/filter) will actually do (p 40).

TABLE 1.23 A query design for selecting all the sales of vegetarian pizzas since the start of February. It is sorted by CustID type (primary sort key), then by Order Date (secondary sort key) and Order Time (tertiary sort key). There is a formula to calculate the total price for each transaction. All the pizzas are vegetarian so the PizzaName field will not be displayed in the final list because the data would be redundant. Note that this query uses fields drawn from more than one table.

Field	tblOrders Items	tblOrder Items	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders
tblOrderID	tblOrderID	tblOrderID	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders
tblCustomerID	tblCustomerID	tblCustomerID	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders
tblOrderDate	tblOrderDate	tblOrderDate	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders
tblOrderTime	tblOrderTime	tblOrderTime	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders
tblPizzas Name	tblPizzas Name	tblPizzas Name	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders
tblPizzas Price	tblPizzas Price	tblPizzas Price	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders
tblPizzas Quantity	tblPizzas Quantity	tblPizzas Quantity	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders
tblPizzas Total	tblPizzas Total	tblPizzas Total	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders
tblPizzas Formula	tblPizzas Formula	tblPizzas Formula	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders	tblPizzas Orders

Query Name: qryVegetarianSalesByCustID Source: tblOrders, tblPizzas, tblPizzas Orders, tblPizzas Orders

Design Tools

- A **Layout diagrams and mockups** shows what an input (form) and output (report) will look like.

Validation techniques

- Manual & electronic validation
- Electronic validation relies on s/w to perform checks on accuracy, completeness and reasonableness
- Range checks, spell checks, grammar checking, predefined lists, data type checks, input masks, alignment and IIF () statements.
- Input mask, "9" allows single digit no and "L" allows single text

Test Data

- Test data is also prepared at the **design** stage ready to be used later in the problem-solving process. A test plan is also developed here.
- This data should be prepared on the basis of **ensuring** the solution functions correctly (p 41).
- The test data should test that validation rules (range checks, input masks, existence checks etc.) are picking up unreasonable or incorrect data, that calculations are returning the correct result, that queries filter out the correct data and that macros complete the tasks required.

Validation techniques

Field Name	Data Type	Field Properties
cusCustomerID	Text	Customer's individual assigned code
cusFamilyName	Text	Customer's family name
cusGivenName	Text	Customer's given name
cusDelAddress	Text	Customer's delivery address
cusDelSuburb	Text	Customer's delivery suburb
cusDelPostCode	Text	Customer's delivery postcode
cusMobile	Text	Customer's mobile number
cusEmail	Text	Customer's email address
cusClub	Yes/No	Is customer a member of the Pizza Club?

General / Lookup	
Field Size	4
Format	
Input Mask	
Caption	Customer ID
Default Value	Between 200 And 9999
Validation Rule	Customer ID number must be between 200 and 9999 inclusive.

FIGURE 1.46 A table design screen with a validation rule applied to one of the fields

Test Data

TABLE 1.27 An example of a test plan for a range of common database elements

Form/function to test	Location	Test data	Expected/required result	Why?
Validation on field of cdey (Existence check)	tblOrderItems table	Null value	Null value. An error message about missing data must be displayed.	There is no value entered and the field requires one
		1	No error message will be displayed.	There is data in the field, as it requires.
Validation on field of qty (Data type check)	tblOrderItems table	1	No error message will be displayed.	The field requires a numeric value to be entered and 1 is a numeric value.
		A	An error message will be displayed to indicate that the data type is incorrect because the field requires a numeric value and A is not numeric.	The field requires a numeric value to be entered, but A is a non-numeric value.
Validation on field of qty (Range check)	tblOrderItems table	Order Qty 1 2 9 10 11	No error message will be displayed.	All values are within the range of 1 to 10 inclusive.
		0 11	The message 'Between 1 and 10 (inclusive) pizzas can be ordered' will be displayed.	Both values are outside the range of 1 to 10 inclusive.

Developing a RDBMS

- Creation of RDBMS occurs in development stage of PSM commencing with creation of tables
- Relationships are created, ensuring referential integrity, (presence of data in foreign keys) is applied
- Queries are created

FIGURE 1.55 This query will select all vegetarian pizzas sold since the start of February and sort them by crust type, then order date and order time

Testing the Solution

Testing is an important step in problem solving that occurs during the development stage.

It is performed to ensure that:

- all features and functions work correctly (functionality).
- the information produced is free of error (accuracy).
- The information meets the needs of the user (acceptance testing).

Testing the Solution

- Testing can either occur as the solution is being developed (**informal testing**) or after the overall solution has been developed (**formal testing**).
- A bench test forms part of the formal testing process, using sample data developed during design stage; using test plan created
- Conduct user acceptance testing, ensuring solution meets needs of intended users, asking users to follow a series of steps to complete a task in the solution or interpreting the information in the output
- Both acceptable and unacceptable test data should be used

Test Plan

The test plan is used to allow a systematic check of each part of the solution to ensure its proper operation. It involves six steps (three at design stage & three at development stage):

1. Decide on tests to be conducted (design stage).
2. Determine the test data (design stage).
3. Determine expected results (design stage).
4. Record results of testing (develop stage).
5. Correct any errors (develop stage).
6. Provide concluding statement (develop stage).

What to test for.

- Tables, validation, input masks; the validation rules work correctly. With values, test value before the value itself, and one value greater than value in validation rule

FIGURE 1.65 A data type check on the Qty field in the drinks table. The green highlighted areas indicate a validation error – a non-numeric value has been entered. The yellow highlighted areas indicate a numeric value has been correctly entered.

What to test for.

- Queries return the correct result.

FIGURE 1.71 Testing the Count function in the vegetarian pizza query

Function to be tested: Count on PizzaType field in qryVegetarianPizzas query

Test data: records are selected when the query is run

Expected results and reason: The count should be 5 because 5 records are selected when the query is run (refer to highlighted section of Figure 1.71)

Actual result: The count is 5. Therefore, actual result = expected result. ✓ Count OK

What to test for.

- Testing formulas and summary statistics
- Testing sorting
- Macros complete the correct steps.
- Reports follow acceptable conventions.
- Whether the solution meets the needs of the user (user acceptance), especially the output.

Why keep data secure?

We have covered this earlier in the year. All of these reasons can impact on the financial performance of an organisation.

- Competitive advantage.
- Business reputation.
- Damage or loss of data vital to functioning of organisation.
- To maintain public's confidence.
- Privacy legislation.

Keeping data secure

The major benefit of keeping data secure is that it ensures data integrity.

Without data integrity the quality of the information from a database will be lessened (GIGO).

Threats to data can come from a number of sources:

- Deliberate threats
- Accidental threats
- Event-based threats

Security measures, physical

- Physical equipment controls include zoned security strategies, barrier techniques and biometrics.
- Biometrics, a form of identification and access control. Biometrics uses distinctive, measurable characteristics to describe individuals, fingerprint, face recognition, palm print, iris recognition, etc.
- Unique to individuals, so reliable

Security measures, physical & software based

Backing up

- Involves creating a copy of the file in case the original is lost or damaged.
- The timing, location, extent and frequency of the backing up are important consideration.
- Full, differential and incremental backups
- Various backup medium & media, including cloud storage, portable devices, etc.

Security measures

Electrical protection

Protection against power events such as stoppages or power surges is vital.

Back up power supplies and batteries can protect against outages. An uninterruptible power supply (UPS) can regulate electrical surges and allow short term power for backing up if a blackout occurs.

Security measures

Username and passwords

- These are a common way of restricting access to data.
- Levels of access to a database can be regulated, for instance who is able to read and who is able to write to the database.
- Access can be monitored through audits and logs; use of access restrictions
- Rules relating to passwords should be followed.

Security measures

System security software

- Various types of malicious software (malware) & phishing can impact on data integrity.
- There is a variety of security software available, although often at a cost.
- Role of firewalls
- Important that the security software is constantly updated to allow for changes in malware.

Security measures: Encryption

- This involves encoding the data to prevent interception or theft. If unauthorised access occurs the data cannot be read.
- This can be used both when transmitting data or storing data.
- Examples of use with a website include SSL, TLS or using a HTTPS connection.
- Encrypting can reduce efficiency because of the extra time taken.
- Hashing, contents of databases encrypted inside fields